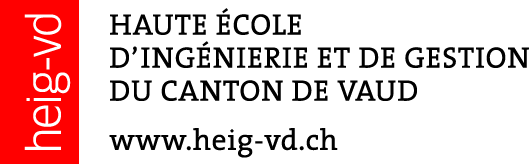
June 2017

Laboratory Report

HTTP Infrastructure

RES

Prof. Olivier Liechti

Bichon Passuello

Max Caduff

Step 1:

The objective of this first part is to implement a static HTTP Apache server, using existing Docker images.

We used the Docker image « PHP » (<https://hub.docker.com/_/php/>) as a base, and the template « Stylish Portfolio » from startboostrap (<https://startbootstrap.com/template-overviews/stylish-portfolio/>) slightly modified as the content to be displayed. The website is located in the content/ folder, which is copied in the var/www/html/ folder of the Container.

The DockerFile to do this is as follows:

FROM php:7.0-apache

COPY content/ var/www/html/

To build the image, we run « sudo docker build -t res/apache\_php . » from the folder containing

content/ and the Dockerfile. (apache-php-image).

Then, to run the container, we should give it the local port to communicate with it (here 9090) and the port of the container on which it should listen (here 80). Hence the command: « sudo docker run -p 9090:80 res/apache\_php »

In order to intact with the machine, we can launch a bash instance when running it with:

« sudo docker run -i -p 9090:80 res/apache\_php /bin/bash »

we can then explore the filesystem and find the location of the Apache config files, here:

/var/lib/apache2/conf

Step 2:

In this part we needed to implement a dynamic server from the same Docker image as Step 1, using node.js to generate the response, and express.js to manage the server behavior.

The server generates a fake Finnish dialog made of a random number of random sentences, the first is a 5-word question, the answer has an unbounded number of words.

The DockerFile is as follows:

FROM node:6.10

COPY src /opt/app

CMD ["node", « opt/app/index.js"]

The command to build the image is: « sudo docker build -t res/express\_students . »

The command to run the container is: « sudo docker run -p 9090:3000 res/express\_students »

We need once again to map the correct ports of the machine, since the container is listening on his port 3000.

Step 3

Still using the PHP 7.0 image, the reverse proxy virtual hosts are configured as follow :

The 001-reverse-proxy.conf file is as follow :

VirtualHost \*:80>

ServerName demo.res.ch

#ErrorLog ${APACHE\_LOG\_DIR}/error.log

#CustomLog ${APACHE\_LOG\_DIR}/access.log combined

ProxyPass "/api/finnish/" "http://172.17.0.3:3000/"

ProxyPassReverse "/api/finnish/" "http://172.17.0.3:3000/"

ProxyPass "/" "http://172.17.0.4:80/"

ProxyPassReverse "/" "http://172.17.0.4:80/"

</VirtualHost>

Any request for the uri /api/finnish will redirect to the express\_dynamic vm, and any / GET request will be redirected to the apache\_static vm.

run static server : sudo docker run -d --name apache\_static res/apache\_php

ip address is :

"IPAddress": "172.17.0.4",

run dynamic server : sudo docker run -d --name express\_dynamic res/express\_students

ip address is :

"IPAddress": "172.17.0.3",

Step 4 :

Need to add this to the hosts file : 0.0.0.0 demo.res.ch

172.17.0.2 is the ip for apache\_static vhost

172.17.0.3is the ip for express\_dynamic vhost

Added command to DockerFile to update apt-get and install vim

The js command result replaces the about section in the static page by a finnish sentence every 2 seconds.

Everything else is the same as in the last step